Remarks

In the present response, claims 1-20 are presented for examination.

Claim Rejections: 35 USC § 102(b)

Claims 1, 3, and 12-14 are rejected under 35 USC § 102(b) as being anticipated by "The Design and Implementation of JaDiSM" (Dunn). These rejections are traversed.

The claims recite elements not taught or even suggested in Dunn. Some examples are provided below for the independent claims.

<u>Claim 1</u>

As one example, claim 1 recites first and second page tables that each translate a virtual memory address to different physical addresses. This recitation is fully supported in paragraphs [0016] – [0018] and Fig. 2 in Applicants' specification. Dunn does not teach or even suggest this element.

Dunn teaches a Java library having multiple Java applications distributed across a multiprocessor system. Nodes in the system communicate with each other using UDP (a message protocol for sending short messages to each other over the network). Every object in the shared address space is visible across the distributed system memory. When a node wants to share an object, the node publishes a reference to the object to the other nodes. Dunn provides an example wherein a pointer to an object called "foo" on one machine will have the same meaning on another machine since the object foo is shared.

The teachings in Dunn are quite different than the elements in claim 1. For example, nowhere does Dunn teach or even suggest that the nodes in his system have page tables that translate one virtual memory address to different physical addresses.

Dunn does teach that each node creates and initializes a page-table in its private memory space in order to manage the shared region (see Dunn second paragraph on p. 6). Dunn explains how his page-tables work:

> This page-table consists of a page-table entry for each page in the region whether the node has ownership of the page or not. The page-table entry maintains information about whether the node has permission to read

> or write the page and what node owns the page should it need to request access; it also has the epoch number in which the page was last modified. The owner of a page will have the copyset, or a bit-map of the nodes that share this page. Nodes that write a page maintain information about a twin and a *diff* in the page table. A twin is a copy of the page and is used for creating *diffs*, or a buffer of the modifications made to the page. (See Dunn at p. 6, second paragraph).

Notice that nowhere does Dunn teach or even suggest that his page-tables translate one virtual memory address to different physical addresses.

The differences between claim 1 and Dunn are great since Dunn fails to teach or suggest all of the claim elements. As such, Dunn does not anticipate claim 1.

For at least these reasons, claim 1 and its dependent claims are allowable over the art of record.

<u>Claim 12</u>

As one example, claim 12 recites the following elements:

> addressing data at a same virtual address by different processors in different functional units, wherein each processor in a different functional unit reads different data specific to its functional unit for the same virtual address; and
>
> addressing data at a second same virtual address by the different processors in the different functional units, wherein the second same virtual address in each of the different functional units points to a same physical address.

These recitations are fully supported in paragraphs [0016] – [0018] and Fig. 2 in Applicants' specification. Dunn does not teach or even suggest these elements.

Dunn teaches a Java library having multiple Java applications distributed across a multiprocessor system. Nodes in the system communicate with each other using UDP (a

message protocol for sending short messages to each other over the network). Every object in the shared address space is visible across the distributed system memory. When a node wants to share an object, the node publishes a reference to the object to the other nodes. Dunn provides an example wherein a pointer to an object called "foo" on one machine will have the same meaning on another machine since the object foo is shared.

The teachings in Dunn are quite different than the elements in claim 12. For example, nowhere does Dunn teach or even suggest that the nodes in his system address data at a same virtual address to read different data and then address data at another virtual address to point to the same physical address.

Dunn does teach that each node creates and initializes a page-table in its private memory space in order to manage the shared region (see Dunn second paragraph on p. 6). Dunn explains how his page-tables work:

> This page-table consists of a page-table entry for each page in the region whether the node has ownership of the page or not. The page-table entry maintains information about whether the node has permission to read or write the page and what node owns the page should it need to request access; it also has the epoch number in which the page was last modified. The owner of a page will have the copyset, or a bit-map of the nodes that share this page. Nodes that write a page maintain information about a twin and a *diff* in the page table. A twin is a copy of the page and is used for creating *diffs*, or a buffer of the modifications made to the page. (See Dunn at p. 6, second paragraph).

Notice that nowhere does Dunn teach or even suggest that his page-tables address data at a same virtual address to read different data and then address data at another virtual address to point to the same physical address.

The differences between claim 12 and Dunn are great since Dunn fails to teach or suggest all of the claim elements. As such, Dunn does not anticipate claim 12.

For at least these reasons, claim 12 and its dependent claims are allowable over the art of record.

Claim Rejections: 35 USC § 103(a)

Claim 5 is rejected under 35 USC § 103(a) as being unpatentable over Dunn in view of US publication number 2005/0050191 (Hubis). These rejections are traversed.

As explained above, Dunn does not teach or suggest all of the elements of independent claim 1. Hubis fails to cure these deficiencies. Thus for at least the reasons provided with respect to independent claim 1, dependent claim 5 is allowable over Dunn in view of Hubis.

Claim Rejections: 35 USC § 103(a)

Claims 2, 6-11, and 15-20 are rejected under 35 USC § 103(a) as being unpatentable over Dunn in view of USPN 6,092,157 (Suzuki). These rejections are traversed.

The claims recite one or more elements that are not taught or suggested in Dunn in view of Suzuki. These missing elements show that the differences between the combined teachings in the art and the recitations in the claims are great. As such, the pending claims are <u>not</u> a predictable variation of the art to one of ordinary skill in the art. Some examples are provided below for the independent claim 6, which is selected for discussion.

As one example, independent clam 6 recites the following elements (added from dependent claim 7):

> a RAD specific attribute of the first RAD along with a replicated portion of an operating system stored in the first RAM, wherein the replicated portion of the operating system, when executing in the first RAD, reads the RAD specific attribute of the first RAD by reference to a virtual memory address; and
>
> a RAD specific attribute of the second RAD along with a replicated portion of the operating system stored in the second RAM, wherein the replicated portion of the operating system, when executing in the second

RAD, reads the RAD specific attribute of the second RAD by reference to the virtual memory address.

Dunn in view of Suzuki does not teach or even suggest these elements.

Dunn teaches a Java library having multiple Java applications distributed across a multiprocessor system. Nodes in the system communicate with each other using UDP (a message protocol for sending short messages to each other over the network). Every object in the shared address space is visible across the distributed system memory. When a node wants to share an object, the node publishes a reference to the object to the other nodes. Dunn provides an example wherein a pointer to an object called "foo" on one machine will have the same meaning on another machine since the object foo is shared.

The teachings in Dunn are quite different than the elements in claim 6. For example, nowhere does Dunn teach or even suggest that one node reads a first attribute by reference to a virtual memory address and a second node reads a different attribute by reference to the <u>same</u> virtual address (the virtual address has "the" antecedent basis occurring in the claim between the first and second RAD elements).

Dunn does teach that each node creates and initializes a page-table in its private memory space in order to manage the shared region (see Dunn second paragraph on p. 6). Dunn explains how his page-tables work:

> This page-table consists of a page-table entry for each page in the region whether the node has ownership of the page or not. The page-table entry maintains information about whether the node has permission to read or write the page and what node owns the page should it need to request access; it also has the epoch number in which the page was last modified. The owner of a page will have the copyset, or a bit-map of the nodes that share this page. Nodes that write a page maintain information about a twin and a *diff* in the page table. A twin is a copy of the page and is used for creating *diffs*, or a buffer of the modifications made to the page. (See Dunn at p. 6, second paragraph).

Notice that nowhere does Dunn teach or even suggest that his nodes access page-tables to read a first attribute by reference to a virtual memory address and then read a different attribute by reference to the same virtual address.

The differences between the claims and the teachings in the art are great since the references fail to teach or suggest all of the claim elements. As such, the pending claims are not a predictable variation of the art to one of ordinary skill in the art.

For at least these reasons, independent claims 6 and 15 and their dependent claims are allowable over the art of record.

CONCLUSION

In view of the above, Applicants believe that all pending claims are in condition for allowance. Allowance of these claims is respectfully requested.

Any inquiry regarding this Amendment and Response should be directed to Philip S. Lyren at Telephone No. 832-236-5529. In addition, all correspondence should continue to be directed to the following address:

Hewlett-Packard Company
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

Respectfully submitted,

/Philip S. Lyren #40,709/

Philip S. Lyren
Reg. No. 40,709
Ph: 832-236-5529